I 3

```
   CCCCCCCCCCCC  LLL                 IIIIIIIII     UUU         UUU  TTTTTTTTTTTTTTT  LLL
   CCCCCCCCCCCC  LLL                 IIIIIIIII     UUU         UUU  TTTTTTTTTTTTTTT  LLL
   CCCCCCCCCCCC  LLL                 IIIIIIIII     UUU         UUU  TTTTTTTTTTTTTTT  LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
   CCCCCCCCCCCC  LLLLLLLLLLLLLLL     IIIIIIIII     UUUUUUUUUUUUUUU        TTT        LLLLLLLLLLLLLLL
   CCCCCCCCCCCC  LLLLLLLLLLLLLLL     IIIIIIIII     UUUUUUUUUUUUUUU        TTT        LLLLLLLLLLLLLLL
   CCCCCCCCCCCC  LLLLLLLLLLLLLLL     IIIIIIIII     UUUUUUUUUUUUUUU        TTT        LLLLLLLLLLLLLLL
```

```
**FILE**ID**SHOWSYS                                        F 14





  SSSSSSSS  HH      HH    000000    WW      WW    SSSSSSSS  YY      YY    SSSSSSSS
  SSSSSSSS  HH      HH    000000    WW      WW    SSSSSSSS  YY      YY    SSSSSSSS
SS          HH      HH  00      00  WW      WW  SS          YY      YY  SS
SS          HH      HH  00      00  WW      WW  SS          YY      YY  SS
SS          HH      HH  00      00  WW      WW  SS            YY  YY    SS
SS          HH      HH  00      00  WW      WW  SS            YY  YY    SS
  SSSSSS    HHHHHHHHHH  00      00  WW      WW    SSSSSS        YY        SSSSSS
  SSSSSS    HHHHHHHHHH  00      00  WW      WW    SSSSSS        YY        SSSSSS
        SS  HH      HH  00      00  WW  WW  WW          SS      YY              SS
        SS  HH      HH  00      00  WW  WW  WW          SS      YY              SS
        SS  HH      HH  00      00  WWWW  WWWW          SS      YY              SS      ....
        SS  HH      HH  00      00  WWWW  WWWW          SS      YY              SS      ....
SSSSSSSS    HH      HH    000000    WW      WW  SSSSSSSS        YY      SSSSSSSS        ....
SSSSSSSS    HH      HH    000000    WW      WW  SSSSSSSS        YY      SSSSSSSS        ....


LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0 MODULE showsystem (IDENT = 'V04-000',
0                    ADDRESSING_MODE (EXTERNAL = GENERAL)) =
0
1 BEGIN
1 !
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1 !
1 !
1 !++
1 !
1 ! FACILITY:  SHOW utility
1 !
1 ! ABSTRACT:
1 !       This module contains the routines for the SHOW SYSTEM command
1 !
1 ! ENVIRONMENT:
1 !       VAX native, user and kernel mode
1 !
1 ! AUTHOR:  Gerry Smith           CREATION DATE:  30-Jul-1982
1 !
1 ! MODIFIED BY:
1 !
1 !       V03-007 AEW0002         Anne E. Warner          27-Feb-1984
1 !               Reorganize 'Ph.Mem' format to handle increased process
1 !               working set sizes.
1 !
1 !       V03-006 AEW0001         Anne E. Warner          02-Feb-1984
1 !               Reorganize the SHOW SYSTEM display.
1 !                 - Make the display fit on an 80 character display by
1 !                   taking out the UIC.
1 !                 - Add the qualifier /FULL to display all information
1 !                   plus add a second line with the UIC.
1 !                 - Add the system node name to the header.
1 !                 - Add the number of days to each process CPU time.
1 !                 - Add buffered I/O to the direct I/O for each process.
```

```
 1 !
 1 !      V03-005 LMP0140         L. Mark Pilant,         23-Aug-1983  23:29
 1 !              Add support for alphanumeric UICs.
 1 !
 1 !      V03-004 GAS0117         Gerry Smith             12-Apr-1983
 1 !              Instead of displaying MWAIT, display which resource
 1 !              the process is awaiting, or MUTEX if waiting for that.
 1 !
 1 !      V03-003 CWH1002         CW Hobbs                25-Feb-1983
 1 !              Use extended pids for the process ids.
 1 !
 1 !      V03-002 GAS0107                     Gerry Smith       8-Feb-1983
 1 !              Collect all the qualifiers before making checks on
 1 !              whether or not any were set.
 1 !
 1 !      V03-001 GAS00103                            17-Jan-1983
 1 !              Initialize the PCB before going into the PIX loop.
 1 !
 1 !--
```

```
!
! Include files
!

LIBRARY 'SYS$LIBRARY:LIB';              ! VAX/VMS system definitions
REQUIRE 'SRC$:SHOWDEF';                 ! SHOW common definitions

! Define macro to make a string descriptor:  sd_A

MACRO
    SD[A] =
            BIND %NAME('SD_',A) = $DESCRIPTOR(A)%;

!
! Define the flags for SHOW SYSTEM
!
MACRO
    sys$v_proc  = 0, 0, 1, 0%,
    sys$v_subp  = 0, 1, 1, 0%,
    sys$v_net   = 0, 2, 1, 0%,
    sys$v_batch = 0, 3, 1, 0%,
    sys$v_full  = 0, 4, 1, 0%;

!
! Macros to define the layout of the data block to used to
! hold information on a particular process
!
MACRO
    d$l_pid     =  0, 0, 32, 0%,                ! Process ID
    d$l_owner   =  4, 0, 32, 0%,                ! Temp owner storage/process name length
    d$a_name    =  8, 0, 32, 0%,                ! Pointer to process name string
    d$l_state   = 12, 0, 32, 0%,                ! Process state
    d$l_pri     = 16, 0, 32, 0%,                ! Current priority
    d$l_iocnt   = 20, 0, 32, 0%,                ! Direct plus Buffered I/O count
    d$l_cputim  = 24, 0, 32, 0%,                ! CPU time
    d$l_pflts   = 28, 0, 32, 0%,                ! Page fault count
    d$l_pgcnt   = 32, 0, 32, 0%,                ! Global page count
    d$l_sts     = 36, 0, 32, 0%,                ! Status
    d$l_uic     = 40, 0, 32, 0%,                ! Process UIC
    d$l_lef     = 44, 0, 32, 0%,                ! Local event flag
    d$t_name    = 48, 0,  8, 0%;                ! Process name

!
! The following literal depends on D$T_NAME being the last field in the
! locked data area.
!
LITERAL d$k_length = $BYTEOFFSET(d$t_name) + pcb$s_lname;

!
! Define two bits in the data area, D$V_NETWRK and D$V_BATCH, which
! correspond to PCB$V_NETWORK and PCB$V_BATCH, except that they reference
! the process status from the PCB status longword, rather than from the
! beginning of the PCB.
!
MACRO
    d$v_netwrk = 0, $BITPOSITION(pcb$v_netwrk), $FIELDWIDTH(pcb$v_netwrk), $EXTENSION(pcb$v_netwrk)%,
;  135          0232  1      d$v_batch  = 0, $BITPOSITION(pcb$v_batch),  $FIELDWIDTH(pcb$v_batch),  $EXTENSION(pcb$v_batch)%;
```

```
!
! Construct a table of process states.  THIS IS AN ORDERED TABLE.
!
LITERAL rsn_cnt = 14;
OWN
    state_table : VECTOR[14]
    INITIAL (cstring('COLPG'),
             cstring('MUTEX'),
             cstring('CEF'),
             cstring('PFW'),
             cstring('LEF'),
             cstring('LEFO'),
             cstring('HIB'),
             cstring('HIBO'),
             cstring('SUSP'),
             cstring('SUSPO'),
             cstring('FPG'),
             cstring('COM'),
             cstring('COMO'),
             cstring('CUR')),
    rsn_table : VECTOR[rsn_cnt]
    INITIAL (cstring('RWAST'),                        ! AST wait
             cstring('RWMBX'),                        ! Mailbox full
             cstring('RWNPG'),                        ! Non-paged pool
             cstring('RWPFF'),                        ! Page file full
             cstring('RWPAG'),                        ! Paged pool
             cstring('RWBRK'),                        ! Waiting for BROADCAST to finish
             cstring('RWIMG'),                        ! Image activation lock
             cstring('RWQUO'),                        ! Pooled quota
             cstring('RWLCK'),                        ! Lock ID data base
             cstring('RWSWP'),                        ! Swap file space
             cstring('RWMPE'),                        ! Modified page list empty
             cstring('RWMPB'),                        ! Modified page writer busy
             cstring('RWSCS'),                        ! SCS wait
             cstring('RWCLU'));                       ! Cluster translation wait

!
! This ASSUME macro makes sure that we have all the miscellaneous resource
! waits in this table.
!
$ASSUME(rsn_cnt, EQL, rsn$_max-1)

```

```
1 !
1 ! Table of contents
1 !
1
1 FORWARD ROUTINE
1     show$system : NOVALUE,
1     get_data,
1     print_data : NOVALUE;
1
1 FORWARD
1     lock_start : VECTOR[0],
1     lock_end : VECTOR[0];
1
1 EXTERNAL
1     scs$ga_localsb,                                   ! Local system block
1     sys$gq_version,                                   ! System version
1     exe$gl_abstim,                                    ! Time system has been up
1     sch$gl_pcbvec : REF VECTOR,                       ! PCB vector
1     sch$gl_maxpix;                                    ! Maximum process index count
1
1 EXTERNAL ROUTINE
1     lib$get_vm,
1     cli$present,
1     show$write_line : NOVALUE;
```

```
1 GLOBAL ROUTINE show$system : NOVALUE =
2 BEGIN
2
2 !---
2 !
2 ! This is the driver routine for the SHOW SYSTEM function.  The command
2 ! qualifiers are gathered, scratch space is allocated, the data-gathering
2 ! routine is called via $CMKRNL, and then the data is printed.
2 !
2 !---
2
2 LOCAL
2     status,                                ! General status return
2     size,                                  ! Size of scratch area
2     flags : $BBLOCK[2],                    ! Flags byte
2     desc : VECTOR[2],                      ! Argument list for calls
2     data : VECTOR[2];                      ! Address limits of scratch area
2
2 !
2 ! Collect qualifiers.
2 !
2 flags[sys$v_full]  = cli$present(%ASCID 'FULL');
2 flags[sys$v_subp]  = cli$present(%ASCID 'SUBPROCESS');
2 flags[sys$v_net]   = cli$present(%ASCID 'NETWORK');
2 flags[sys$v_batch] = cli$present(%ASCID 'BATCH');
3 IF NOT (.flags[sys$v_subp] OR
3         .flags[sys$v_net] OR
3         .flags[sys$v_batch])
2 THEN flags[sys$v_proc] = true;
2
2 !
2 ! Allocate a scratch area in which to put data about the processes.
2 ! The size of the scratch area is determined by taking the amount of
2 ! bytes of information per process (D$K_LENGTH), multiplying that by
2 ! the maximum number of processes in the system, and then adding a few
2 ! pages for slop.  The beginning and ending addresses of the area will
2 ! be returned in DATA.
2 !
2 size = (.sch$gl_maxpix * d$k_length) + (3 * 512);
2
3 IF NOT (status = LIB$GET_VM(size,                   ! This many bytes
3                             data))                  ! Put starting address here
2 THEN SIGNAL_STOP(show$_insvirmem, 0, .status);      ! Stop if error
2 data[1] = .data[0] + .size - 1;                     ! Put ending address here
2
2 !
2 ! Lock the first page of the scratch area, and the code that runs at elevated
2 ! IPL, into the process working set.
2 !
2 desc[0] = lock_start;
2 desc[1] = lock_end;
3 IF NOT (status = $LKWSET(INADR = desc))
2 THEN SIGNAL_STOP(.status);
2
2 desc[0] = .data[0];
2 desc[1] = .data[0] + d$k_length;
3 IF NOT (status = $LKWSET(INADR = desc))
```

```
   2 THEN SIGNAL_STOP(.status);
   2
   2 !
   2 ! Call the data-gathering routine in kernel mode, passing the address
   2 ! limits as an argument.
   2 !
   2 desc[0] = 2;
   2 desc[1] = data;
   2 desc[2] = flags;
P  3 IF NOT (status = $CMKRNL(ROUTIN = get_data,
   3                          ARGLST = desc))
   2 THEN
   3     BEGIN
   3     SIGNAL(.status);
   3     RETURN;
   2     END;
   2
   2 !
   2 ! Format and print the data.
   2 !
   2 print_data(data, flags);
   2
   2 RETURN;
   1 END;                                                    ! End of show$error

                                                              .TITLE  SHOWSYSTEM
                                                              .IDENT  \V04-000\

                                                              .PSECT  $PLIT$,NOWRT,NOEXE,2

       47  50  4C  4F  43  05  00000 P.AAA:  .ASCII  <5>\COLPG\                          ;
       58  45  54  55  4D  05  00006 P.AAB:  .ASCII  <5>\MUTEX\                          ;
               46  45  43  03  0000C P.AAC:  .ASCII  <3>\CEF\                            ;
               57  46  50  03  00010 P.AAD:  .ASCII  <3>\PFW\                            ;
               46  45  4C  03  00014 P.AAE:  .ASCII  <3>\LEF\                            ;
           4F  46  45  4C  04  00018 P.AAF:  .ASCII  <4>\LEFO\                           ;
               42  49  48  03  0001D P.AAG:  .ASCII  <3>\HIB\                            ;
           4F  42  49  48  04  00021 P.AAH:  .ASCII  <4>\HIBO\                           ;
           50  53  55  53  04  00026 P.AAI:  .ASCII  <4>\SUSP\                           ;
       4F  50  53  55  53  05  0002B P.AAJ:  .ASCII  <5>\SUSPO\                          ;
               47  50  46  03  00031 P.AAK:  .ASCII  <3>\FPG\                            ;
               4D  4F  43  03  00035 P.AAL:  .ASCII  <3>\COM\                            ;
           4F  4D  4F  43  04  00039 P.AAM:  .ASCII  <4>\COMO\                           ;
               52  55  43  03  0003E P.AAN:  .ASCII  <3>\CUR\                            ;
       54  53  41  57  52  05  00042 P.AAO:  .ASCII  <5>\RWAST\                          ;
       58  42  4D  57  52  05  00048 P.AAP:  .ASCII  <5>\RWMBX\                          ;
       47  50  4E  57  52  05  0004E P.AAQ:  .ASCII  <5>\RWNPG\                          ;
       46  46  50  57  52  05  00054 P.AAR:  .ASCII  <5>\RWPFF\                          ;
       47  41  50  57  52  05  0005A P.AAS:  .ASCII  <5>\RWPAG\                          ;
       4B  52  42  57  52  05  00060 P.AAT:  .ASCII  <5>\RWBRK\                          ;
       47  4D  49  57  52  05  00066 P.AAU:  .ASCII  <5>\RWIMG\                          ;
       4F  55  51  57  52  05  0006C P.AAV:  .ASCII  <5>\RWQUO\                          ;
       4B  43  4C  57  52  05  00072 P.AAW:  .ASCII  <5>\RWLCK\                          ;
       50  57  53  57  52  05  00078 P.AAX:  .ASCII  <5>\RWSWP\                          ;
       45  50  4D  57  52  05  0007E P.AAY:  .ASCII  <5>\RWMPE\                          ;
       42  50  4D  57  52  05  00084 P.AAZ:  .ASCII  <5>\RWMPB\                          ;
```

```
                              53  43  53  57  52  05  0008A P.ABA:   .ASCII   <5>\RWSCS\                                    ;
                              55  4C  43  57  52  05  00090 P.ABB:   .ASCII   <5>\RWCLU\                                    ;
                                                      00096          .BLKB    2
                                      4C  4C  55  46  00098 P.ABD:   .ASCII   \FULL\                                        ;
                                            010E0004  0009C P.ABC:   .LONG    17694724                                      ;
                                           00000000'  000A0          .ADDRESS P.ABD                                         ;
          00  00  53  53  45  43  4F  52  50  42  55  53  000A4 P.ABF:   .ASCII   \SUBPROCESS\<0><0>                            ;
                                            010E000A  000B0 P.ABE:   .LONG    17694730                                      ;
                                           00000000'  000B4          .ADDRESS P.ABF                                         ;
                      00  4B  52  4F  57  54  45  4E  000B8 P.ABH:   .ASCII   \NETWORK\<0>                                  ;
                                            010E0007  000C0 P.ABG:   .LONG    17694727                                      ;
                                           00000000'  000C4          .ADDRESS P.ABH                                         ;
                      00  00  00  48  43  54  41  42  000C8 P.ABJ:   .ASCII   \BATCH\<0><0><0>                              ;
                                            010E0005  000D0 P.ABI:   .LONG    17694725                                      ;
                                           00000000'  000D4          .ADDRESS P.ABJ                                         ;

                                                                     .PSECT   $OWN$,NOEXE,2

00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00000 STATE_TABLE:
                                                                     .ADDRESS P.AAA, P.AAB, P.AAC, P.AAD, P.AAE, -           ;
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00018             P.AAF, P.AAG, P.AAH, P.AAI, P.AAJ, P.AAK, -    ;
                                        00000000' 00000000' 00030             P.AAL, P.AAM, P.AAN                            ;
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00038 RSN_TABLE:
                                                                     .ADDRESS P.AAO, P.AAP, P.AAQ, P.AAR, P.AAS, -           ;
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00050             P.AAT, P.AAU, P.AAV, P.AAW, P.AAX, P.AAY, -    ;
                                        00000000' 00000000' 00068             P.AAZ, P.ABA, P.ABB                            ;

                                                                     .EXTRN   SCS$GA_LOCALSB, SYS$GQ_VERSION
                                                                     .EXTRN   EXE$GL_ABSTIM, SCH$GL_PCBVEC
                                                                     .EXTRN   SCH$GL_MAXPIX, LIB$GET_VM
                                                                     .EXTRN   CLI$PRESENT, SHOW$WRITE_LINE
                                                                     .EXTRN   SYS$LKWSET, SYS$CMKRNL

                                                                     .PSECT   $CODE$,NOWRT,2

                                                  003C  00000        .ENTRY   SHOW$SYSTEM, Save R2,R3,R4,R5                 ; 0299
                         55  00000000G   00   9E  00002              MOVAB    SYS$LKWSET, R5                                ;
                         54  00000000G   00   9E  00009              MOVAB    LIB$STOP, R4                                  ;
                         53  00000000G   00   9E  00010              MOVAB    CLI$PRESENT, R3                               ;
                         5E              18   C2  00017              SUBL2    #24, SP                                       ;
                                  0000'  CF   9F  0001A              PUSHAB   P.ABC                                         ; 0320
                         63              01   FB  0001E              CALLS    #1, CLI$PRESENT                               ;
     04   AE        01   04              50   F0  00021              INSV     R0, #4, #1, FLAGS                             ;
                                  0000'  CF   9F  00027              PUSHAB   P.ABE                                         ; 0321
                         63              01   FB  0002B              CALLS    #1, CLI$PRESENT                               ;
     04   AE        01   01              50   F0  0002E              INSV     R0, #1, #1, FLAGS                             ;
                                  0000'  CF   9F  00034              PUSHAB   P.ABG                                         ; 0322
                         63              01   FB  00038              CALLS    #1, CLI$PRESENT                               ;
     04   AE        01   02              50   F0  0003B              INSV     R0, #2, #1, FLAGS                             ;
                                  0000'  CF   9F  00041              PUSHAB   P.ABI                                         ; 0323
                         63              01   FB  00045              CALLS    #1, CLI$PRESENT                               ;
     04   AE        01   03              50   F0  00048              INSV     R0, #3, #1, FLAGS                             ;
                    0E        04   AE    01   E0  0004E              BBS      #1, FLAGS, 1$                                 ; 0324
                    09        04   AE    02   E0  00053              BBS      #2, FLAGS, 1$                                 ; 0325
                    04        04   AE    03   E0  00058              BBS      #3, FLAGS, 1$                                 ; 0326
                              04   AE    01   88  0005D              BISB2    #1, FLAGS                                     ; 0327
                    50 00000000G   00    06   78  00061 1$:          ASHL     #6, SCH$GL_MAXP'X, R0                         ; 0337
```

```
                                6E      0600  C0  9E 00069        MOVAB   1536(R0), SIZE                         ;
                                          08  AE  9F 0006E        PUSHAB  DATA                                   ; 0339
                                          04  AE  9F 00071        PUSHAB  SIZE                                   ;
                       00000000G  00          02  FB 00074        CALLS   #2, LIB$GET_VM                         ;
                                  52          50  D0 0007B        MOVL    R0, STATUS                             ;
                                  0D          52  E8 0007E        BLBS    STATUS, 2$                             ;
                                              52  DD 00081        PUSHL   STATUS                                 ; 0341
                                              7E  D4 00083        CLRL    -(SP)                                  ;
                                    007812F2  8F  DD 00085        PUSHL   #7869170                               ;
                                  64          03  FB 0008B        CALLS   #3, LIB$STOP                           ;
                 50          08   AE          6E  C1 0008E 2$:    ADDL3   SIZE, DATA, R0                         ; 0342
                             0C   AE          FF  A0  9E 00093    MOVAB   -1(R0), DATA+4                         ;
                             10   AE       0000V  CF  9E 00098    MOVAB   LOCK_START, DESC                       ; 0348
                             14   AE       0000V  CF  9E 0009E    MOVAB   LOCK_END, DESC+4                       ; 0349
                                              7E  7C 000A4        CLRQ    -(SP)                                  ; 0350
                                          18  AE  9F 000A6        PUSHAB  DESC                                   ;
                                  65          03  FB 000A9        CALLS   #3, SYS$LKWSET                         ;
                                  52          50  D0 000AC        MOVL    R0, STATUS                             ;
                                  05          52  E8 000AF        BLBS    STATUS, 3$                             ;
                                              52  DD 000B2        PUSHL   STATUS                                 ; 0351
                                  64          01  FB 000B4        CALLS   #1, LIB$STOP                           ;
                             10   AE          08  AE  D0 000B7 3$: MOVL   DATA, DESC                             ; 0353
           14    AE          08   AE 00000040  8F  C1 000BC       ADDL3   #64, DATA, DESC+4                      ; 0354
                                              7E  7C 000C6        CLRQ    -(SP)                                  ; 0355
                                          18  AE  9F 000C8        PUSHAB  DESC                                   ;
                                  65          03  FB 000CB        CALLS   #3, SYS$LKWSET                         ;
                                  52          50  D0 000CE        MOVL    R0, STATUS                             ;
                                  05          52  E8 000D1        BLBS    STATUS, 4$                             ;
                                              52  DD 000D4        PUSHL   STATUS                                 ; 0356
                                  64          01  FB 000D6        CALLS   #1, LIB$STOP                           ;
                             10   AE          02  D0 000D9 4$:    MOVL    #2, DESC                               ; 0362
                             14   AE          08  AE  9E 000DD    MOVAB   DATA, DESC+4                           ; 0363
                                  6D          04  AE  9E 000E2    MOVAB   FLAGS, DESC+8                          ; 0364
                                          10  AE  9F 000E6        PUSHAB  DESC                                   ; 0366
                                       0000V  CF  9F 000E9        PUSHAB  GET_DATA                               ;
                       00000000G  00          02  FB 000ED        CALLS   #2, SYS$CMKRNL                         ;
                                  52          50  D0 000F4        MOVL    R0, STATUS                             ;
                                  0A          52  E8 000F7        BLBS    STATUS, 5$                             ;
                                              52  DD 000FA        PUSHL   STATUS                                 ; 0369
                       00000000G  00          01  FB 000FC        CALLS   #1, LIB$SIGNAL                         ;
                                                  04 00103        RET                                            ; 0368
                                          04  AE  9F 00104 5$:    PUSHAB  FLAGS                                  ; 0376
                                          0C  AE  9F 00107        PUSHAB  DATA                                   ;
                           0000V  CF          02  FB 0010A        CALLS   #2, PRINT_DATA                         ;
                                                  04 0010F        RET                                            ; 0379

; Routine Size:  272 bytes,    Routine Base:  $CODE$ + 0000
```

```
1 OWN lock_start : VECTOR[0] PSECT ($CODE$);      ! Beginning of locked code
1 ROUTINE get_data (data, flags) =
2 BEGIN
2
2 !---
2 !
2 ! This routine executes in KERNEL mode.  It scans all the processes in the
2 ! system, gathering information on them.
2 !
2 ! Inputs
2 !       DATA  -- address of the scratch area
2 !       FLAGS -- options longword, to tell what kind of processes are desired
2 !
2 ! Outputs
2 !       DATA -- will contain information on the processes
2 !
2 !--
2
2 MAP
2     data : REF VECTOR,
2     flags : REF $BBLOCK;
2
2 REGISTER
2     locked : REF $BBLOCK,                          ! Pointer to locked page
2     scratch : REF $BBLOCK,                         ! Pointer to scratch area
2     pcb : REF $BBLOCK,                             ! Pointer to PCB
2     null,                                          ! Null process PCB address
2     pix;                                           ! Process index
2
2 !
2 ! The first page of the scratch area is locked, so that it can be accessed at
2 ! elevated IPL.  This locked portion will be a temporary storage place for
2 ! information about one process at a time.  The remainder of the scratch area
2 ! will contain information on the processes which are to be displayed.
2 ! Set up these areas so that they can be addressed easily.
2 !
2 locked = .data[0];                                 ! Point to locked area
2 scratch = .data[0] + d$k_length;                   ! Scratch area is just beyond
2                                                    ! the locked data
2
2 null = pcb = .sch$gl_pcbvec[0];                    ! Save address of NULL PCB
2
2 INCR pix FROM 0 TO .sch$gl_maxpix
2 DO
3     BEGIN
3     SET_IPL(IPL$_SYNCH);                           ! Raise IPL
3     IF .pix EQL 0
3     OR (pcb = .sch$gl_pcbvec[.pix]) NEQ .null
3     THEN
4         BEGIN
4         locked[d$l_pid] = .pcb[pcb$l_epid];        ! Use the extended pid
4         locked[d$l_owner] = .pcb[pcb$l_owner];
4         locked[d$l_uic] = .pcb[pcb$l_uic];
4         locked[d$l_state] = .pcb[pcb$w_state];
4         locked[d$l_pri] = .pcb[pcb$b_pri];
4         locked[d$l_pgcnt] = .pcb[pcb$w_ppgcnt] + .pcb[pcb$w_gpgcnt];
4         locked[d$l_lef] = .pcb[pcb$l_efwm];
```

```
;   344        0437  5            IF (locked[d$l_sts] = .pcb[pcb$l_sts])
;   345        0438  4            THEN
;   346        0439  5               BEGIN
;   347        0440  5               locked[d$l_iocnt] = .$BBLOCK[.pcb[pcb$l_phd], phd$l_diocnt] +
;   348        0441  5                                   .$BBLOCK[.pcb[pcb$l_phd], phd$l_biocnt];
;   349        0442  5               locked[d$l_pflts] = .$BBLOCK[.pcb[pcb$l_phd], phd$l_pageflts];
;   350        0443  5               locked[d$l_cputim] = .$BBLOCK[.pcb[pcb$l_phd], phd$l_cputim];
;   351        0444  4               END;
;   352        0445  4            CH$MOVE(pcb$s_lname,
;   353        0446  4                    pcb[pcb$t_lname],
;   354        0447  4                    locked[d$t_name]);
;   355        0448  4            SET_IPL(0);
;   356        0449  4
;   357        0450  4            IF .flags[sys$v_proc]
;   358        0451  5            OR (.flags[sys$v_subp] AND .locked[d$l_owner] NEQ 0)
;   359        0452  5            OR (.flags[sys$v_batch] AND .$BBLOCK[locked[d$l_sts], d$v_batch])
;   360        0453  5            OR (.flags[sys$v_net] AND .$BBLOCK[locked[d$l_sts], d$v_netwrk])
;   361        0454  4            THEN scratch = CH$MOVE(d$k_length, .locked, .scratch);
;   362        0455  3            END;
;   363        0456  3
;   364        0457  3        SET_IPL(0);
;   365        0458  2        END;
;   366        0459  2
;   367        0460  2 RETURN 1;                                                  ! Return
;   368        0461  1 END;                                                ! End of GET_DATA


                                               00110 LOCK_START:
                                                               .BLKB    0


                                          0FFC 00000 GET_DATA:
                                                               .WORD    Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11                  ; 0381
                          56          04    BC   D0 00002      MOVL     @DATA, LOCKED                                         ; 0416
          57        04    BC 00000040       8F   C1 00006      ADDL3    #64, @DATA, SCRATCH                                   ; 0417
                          5B 00000000G      00   D0 0000F      MOVL     SCH$GL_PCBVEC, R11                                    ; 0420
                          58                6B   D0 00016      MOVL     (R11), PCB                                            ;
                          59                58   D0 00019      MOVL     PCB, NULL                                             ;
                          5A                01   CE 0001C      MNEGL    #1, PIX                                               ; 0422
                                          0092   31 0001F      BRW      8$                                                    ;
                          12                08   DA 00022 1$:  MTPR     #8, #18                                               ; 0425
                                            5A   D5 00025      TSTL     PIX                                                   ; 0426
                                            09   13 00027      BEQL     2$                                                    ;
                          58              6B4A   D0 00029      MOVL     (R11)[PIX], PCB                                       ; 0427
                          59                58   D1 0002D      CMPL     PCB, NULL                                             ;
                                            7F   13 00030      BEQL     7$                                                    ;
                          66          64    A8   D0 00032 2$:  MOVL     100(PCB), (LOCKED)                                    ; 0430
                    04    A6          1C    A8   D0 00036      MOVL     28(PCB), 4(LOCKED)                                    ; 0431
                    28    A6        00BC    C8   D0 0003B      MOVL     188(PCB), 40(LOCKED)                                  ; 0432
                    0C    A6          2C    A8   3C 00041      MOVZWL   44(PCB), 12(LOCKED)                                   ; 0433
                    10    A6          0B    A8   9A 00046      MOVZBL   11(PCB), 16(LOCKED)                                   ; 0434
                          50          36    A8   3C 0004B      MOVZWL   54(PCB), R0                                           ; 0435
                          51          34    A8   3C 0004F      MOVZWL   52(PCB), R1                                           ;
       20    A6           50                51   C1 00053      ADDL3    R1, R0, 32(LOCKED)                                    ;
                    2C    A6          4C    A8   D0 00058      MOVL     76(PCB), 44(LOCKED)                                   ; 0436
```

```
                                    50        24   A8   D0 0005D          MOVL    36(PCB), R0                              ; 0437
                               24   A6             50   D0 00061          MOVL    R0, 36(LOCKED)                           ;
                                    15             50   E9 00065          BLBC    R0, 3$                                   ;
                                    50        6C   A8   D0 00068          MOVL    108(PCB), R0                             ; 0440
                14     A6      54   A0        58   A0   C1 0006C          ADDL3   88(R0), 84(R0), 20(LOCKED)               ; 0441
                               1C   A6        4C   A0   D0 00073          MOVL    76(R0), 28(LOCKED)                       ; 0442
                               18   A6        38   A0   D0 00078          MOVL    56(R0), 24(LOCKED)                       ; 0443
                30     A6      70   A8             10   28 0007D 3$:      MOVC3   #16, 112(PCB), 48(LOCKED)                ; 0447
                                    12             00   DA 00083          MTPR    #0, #18                                  ; 0448
                                    1E        08   BC   E8 00086          BLBS    @FLAGS, 6$                               ; 0450
                       05      08   BC             01   E1 0008A          BBC     #1, @FLAGS, 4$                           ; 0451
                                              04   A6   D5 0008F          TSTL    4(LOCKED)                                ;
                                    14             12   12 00092          BNEQ    6$                                       ;
                       05      08   BC             03   E1 00094 4$:      BBC     #3, @FLAGS, 5$                           ; 0452
                       0A      25   A6             06   E0 00099          BBS     #6, 37(LOCKED), 6$                       ;
                       0E      08   BC             02   E1 0009E 5$:      BBC     #2, @FLAGS, 7$                           ; 0453
                       09      26   A6             05   E1 000A3          BBC     #5, 38(LOCKED), 7$                       ;
                       67           66      0040   8F   28 000A8 6$:      MOVC3   #64, (LOCKED), (SCRATCH)                 ; 0454
                                    57             53   D0 000AE          MOVL    R3, SCRATCH                              ;
                                    12             00   DA 000B1 7$:      MTPR    #0, #18                                  ; 0457
        FF64           5A           01 00000000G   00   F1 000B4 8$:      ACBL    SCH$GL_MAXPIX, #1, PIX, 1$               ; 0422
                                    50             01   D0 000BE          MOVL    #1, R0                                   ; 0460
                                                        04 000C1          RET                                              ; 0461

; Routine Size:  194 bytes,    Routine Base:  $CODE$ + 0110

;  369          0462  1 OWN lock_end : VECTOR[0] PSECT ($CODE$);          ! End of locked code
```

```
1 ROUTINE print_data (data,flags) : NOVALUE =
2 BEGIN
2
2 !---
2 !
2 ! This routine prints the data contained in DATA, the scratch area
2 !
2 ! Inputs
2 !       DATA -- scratch area, contains the process information (didn't I
2 !               just say that?)
2 !
2 !       FLAGS -- contains the bits set for the qualifiers.  It is specifically
2 !                used in this routine to determine if the /FULL qualifier is
2 !                set.
2 !
2 ! Outputs
2 !       The process data is printed.
2 !
2 !--
2
2 MAP
2     data : REF VECTOR,
2     flags: REF $BBLOCK;
2
2 LOCAL
2     scratch : REF $BBLOCK,                        ! Pointer to scratch area
2     status,                                       ! General status
2     time : VECTOR[2],                             ! Place to put system time
2     desc : VECTOR[2],                             ! Descriptor for $FAOL
2     proctim : VECTOR[4],                          ! Storage area for process time
2     arglist : VECTOR[4],                          ! Argument list for $FAOL
2     sysnodebuf : VECTOR[16,BYTE],                 ! 16 byte buffer to receive system node
2     desc_sysnode: $BBLOCK[DSC$C_S_BLN],           ! String descriptor pointing to
2                                                   ! sysnodebuf
2     trnlnmlst : $ITMLST_DECL (ITEMS = 1);         ! Item list for translating system
2                                                   ! node.  [vms.lib]utldef.b32
2
2 ! Set up string descriptor to find the logical name system table
2
2 sd ('LNM$SYSTEM');
2
2 !
2 ! Set up the scratch area, which contains all the data about the processes.
2 ! The data is located beyond the locked segment.
2 !
2 scratch = .data[0] + d$k_length;                          ! Process data begins here.
2
2 !
2 ! If there is no data in the scratch area, then simply return.
2 !
2 IF .scratch[d$l_pid] EQL 0
2 THEN RETURN;
2
2 !
2 ! Determine the time the system has been up.  This is done using the
2 ! value of EXE$GL_ABSTIM, multiplying it by the right constant and
2 ! handing it to $ASCTIM.
```

```
  2 !
  2 EMUL(%REF(.exe$gl_abstim), %REF(-10000000), %REF(0), time); ! Get the uptime
  2 desc[0] = d$k_length;                             ! Set up a descriptor pointing
  2 desc[1] = .data[0];                               ! to the locked (scratch) area
P 3 IF NOT (status = $ASCTIM(TIMADR = time,           ! Convert the uptime
P 3                          TIMBUF = desc,           ! to ASCII, storing here,
P 3                          TIMLEN = desc,           ! put length here,
  3                          CVTFLG = 0))             ! and give full date and time
  2 THEN
  3     BEGIN
  3     SIGNAL(.status);
  3     RETURN;
  2     END;
  2
  2 ! Initialize the descriptor for system node.
  2
  2 desc_sysnode = 16;                                ! Address of space containing
  2 desc_sysnode[dsc$a_pointer] = sysnodebuf;         ! system node
  2
  2 ! Initialize item list used to find system node
  2
P 2 $ITMLST_INIT (ITMLST = trnlnmlst,                 ! Pre-declared address for item list
P 2               (ITMCOD = LNM$_STRING,              ! Logical name translation string
P 2                                                   ! is to be obtained
P 2                BUFADR = sysnodebuf,               ! Address to put system node
P 2                BUFSIZ = 16,                       ! Length of BUFADR
P 2                RETLEN = desc_sysnode)             ! Length of returned node
  2           );
  2
  2 ! Get system node.  Documented in Specification for VMS Logical Name Extension
  2
P 3 IF NOT (status = $TRNLNM
P 3             (ATTR   = %REF(lnm$m_case_blind),     ! Letter case makes no difference
P 3              TABNAM = SD_LNM$SYSTEM,              ! Logical name table to be searched
P 3              LOGNAM = $DESCRIPTOR('SYS$NODE'),    ! What to translate
P 3              ACMODE = %REF(PSL$C_EXEC),           ! Access mode to use
P 3              ITMLST = trnlnmlst                   ! Predefined item list
  4              )
  3          )
  2     THEN desc_sysnode[dsc$w_length] = 0
  2     ELSE
  2
  2 ! Strip leading underscore and trailing colons, if either, from node name
  3
  3     BEGIN
  3       IF .sysnodebuf[0] EQL '_'
  3       THEN
  4       BEGIN
  4         desc_sysnode[dsc$w_length] = .desc_sysnode[dsc$w_length] - 1;
  4         desc_sysnode[dsc$a_pointer] = .desc_sysnode[dsc$a_pointer] + 1;
  3       END;
  3
  3       INCRU I FROM 0 TO 1
  3         DO IF NOT CH$FAIL(CH$FIND_CH(.desc_sysnode[dsc$w_length],
  3                     .desc_sysnode[dsc$a_pointer], ':'))
  3             THEN desc_sysnode[dsc$w_length] = .desc_sysnode[dsc$w_length] - 1;
  2      END;
```

```
2 !
2 ! Set up the $FAOL parameter list, with the addresses of the descriptors of
2 ! the system version, the system node name, and the uptime.
2 !
2
2 arglist[0] = UPLIT(4, sys$gq_version);           ! Version number is 4 bytes
2 arglist[1] = desc_sysnode;                       ! System node name string desc
2 arglist[2] = 0;                                  ! Zero to get current date
2 arglist[3] = desc;                               ! Uptime string desc
2 desc[0] = .desc[0] - 3;                          ! Get rid of trailing ".00"
2
2 !
2 ! Now format and print the header lines.
2 !
2 show$write_line(%ASCID 'VAX/VMS !AS  on node !AS !%D   Uptime !AS',
2                 arglist,
2                 %ASCID '  Pid    Process Name    State  Pri     I/O       CPU       Page flts Ph.Mem',
2                 0);
2
2 !
2 ! Loop thru the scratch area, formatting and outputting the data one process
2 ! at a time.  The data block for each process is set up in as an ordered
2 ! sequence of longwords, in the order of the arguments to $FAOL, so that the
2 ! data block itself can be used as the parameter list to $FAOL.  All that is
2 ! required is some minor fixup.
2 !
2 WHILE .scratch[d$l_pid] NEQ 0                    ! Loop thru all processes
2 DO
3     BEGIN
3 !
3 ! Get the state.
3 !
3     IF .scratch[d$l_state] GEQ sch$c_colpg
3     AND .scratch[d$l_state] LEQ sch$c_cur
3     THEN
4         BEGIN
4         IF .scratch[d$l_state] EQL sch$c_mwait
4         AND .scratch[d$l_lef] GEQ 0
4         THEN
5             BEGIN
5             IF .scratch[d$l_lef] GEQ 1
5             AND .scratch[d$l_lef] LEQ rsn_cnt
5             THEN scratch[d$l_state] = .rsn_table[.scratch[d$l_lef] - 1]
5             ELSE scratch[d$l_state] = cstring('RWUNK');
5             END
4         ELSE  scratch[d$l_state] = .state_table[.scratch[d$l_state] - 1];
4         END
3     ELSE scratch[d$l_state] = cstring('UNK');
3
3 !
3 ! If the owner field is not empty, then the process is a subprocess;
3 ! otherwise, check the status bits for a network or batch process.
3 !
3     status = .scratch[d$l_sts];                  ! Save the status
3     IF .scratch[d$l_owner] NEQ 0
3     THEN scratch[d$l_sts] = cstring('S')
3     ELSE IF .$BBLOCK[scratch[d$l_sts], d$v_netwrk]
```

```
  3        THEN scratch[d$l_sts] = cstring('N')
  3        ELSE IF .$BBLOCK[scratch[d$l_sts], d$v_batch]
  3        THEN scratch[d$l_sts] = cstring('B')
  3        ELSE scratch[d$l_sts] = cstring(' ');
  3
  3   !
  3   ! Convert the priority from the internal format to the external format.
  3   !
  3        scratch[d$l_pri] = 31 - .scratch[d$l_pri];
  3
  3   !
  3   ! The process name string should be converted from the counted string
  3   ! format used in the PCB to a length/address descriptor, so that $FAO
  3   ! can print non-alphanumerics as periods.  To do this, use the OWNER
  3   ! field in the scratch area as the count, and point to the beginning
  3   ! of the saved string, instead of the count.
  3   !
  3        scratch[d$l_owner] = .scratch[d$t_name];
  3        scratch[d$a_name] = scratch[d$t_name] + 1;
  3
  3   !
  3   ! Multiply the CPU time for the process by -100000, setting it up for
  3   ! conversion into ASCII form.
  3   !
  3        IF .status
  3        THEN
  4            BEGIN
  4            EMUL(scratch[d$l_cputim], %REF(-100000), %REF(0), time);
  4            desc[0] = 16;
  4            desc[1] = proctim;                        ! Desc will contain the address
  4                                                      ! of where the actual time is.
P 5            IF NOT (status = $ASCTIM(TIMADR = time,           ! Convert the uptime
P 5                                TIMBUF = desc,        ! to ASCII, storing here,
P 5                                TIMLEN = desc,        ! put length here,
  5                                CVTFLG = 0))          ! and give full date and time
  4            THEN
  5            BEGIN
  5               SIGNAL(.status);
  5               RETURN;
  5            END
  4            ELSE
  4               scratch[d$l_cputim] = desc;
  3            END;
  3   !
  3   ! Now produce the line of text and output it.
  3   !
  3        IF .status
  3        THEN
  4        BEGIN
  4          IF .flags[sys$v_full]
  4          THEN show$write_line(%ASCID '!8XL !15AF !5AC  !3UB!9UL!AS !9UL  !5UW  !AC!/         !%I',
  4               .scratch)
  4          ELSE show$write_line(%ASCID '!8XL !15AF !5AC  !3UB!9UL!AS !9UL  !5UW  !AC',
  4               .scratch)
  4        END
  3          ELSE
  4          BEGIN
```

```
4           IF .flags[sys$v_full]
4           THEN show$write_line(%ASCID '!8XL !15AF !5AC  !3UB!3(+)       --  swapped  out  --         !5UW  !
4                .scratch)
4           ELSE show$write_line(%ASCID '!8XL !15AF !5AC  !3UB!3(+)       --  swapped  out  --         !5UW  !
4                .scratch)
3         END;
3   !
3   ! Adjust the scratch pointer to point to the next block of process info.
3   !
3       scratch = .scratch + d$k_length;
2       END;
2
2 RETURN;
1 END;                                                         ! End of PRINT_DATA


                                                              001D2           .BLKB   2
                                                              001D4 LOCK_END:
                                                                              .BLKB   0

                                                                              .PSECT  $PLIT$,NOWRT,NOEXE,2

                     4D  45  54  53  59  53  24  4D  4E  4C  000D8 P.ABL:  .ASCII  \LNM$SYSTEM\                           ;
                                                             000E2          .BLKB   2
                                                 0000000A    000E4 P.ABK:  .LONG   10
                                                 00000000'   000E8          .ADDRESS P.ABL
                             45  44  4F  4E  24  53  59  53  000EC P.ABN:  .ASCII  \SYS$NODE\
                                                 00000008    000F4 P.ABM:  .LONG   8
                                                 00000000'   000F8          .ADDRESS P.ABN
                                                 00000004    000FC P.ABO:  .LONG   4
                                                 00000000G   00100          .ADDRESS SYS$GQ_VERSION
6E  6F  20  20  53  41  21  20  53  4D  56  2F  58  41  56   00104 P.ABQ:  .ASCII  \VAX/VMS !AS  on node !AS !%D   Uptime !A\   ;
20  20  44  25  21  20  53  41  21  20  65  64  6F  6E  20   00113
                    41  21  20  65  6D  69  74  70  55  20   00122
                                            00  00  00  53   0012C          .ASCII  \S\<0><0><0>
                                                 010E0029.   00130 P.ABP:  .LONG   17694761
                                                 00000000'   00134          .ADDRESS P.ABQ
73  65  63  6F  72  50  20  20  20  20  64  69  50  20  20   00138 P.ABS:  .ASCII  \  Pid    Process Name    State  Pri     \   ;
65  74  61  74  53  20  20  20  20  65  6D  61  4E  20  73   00147
                    20  20  20  20  20  69  72  50  20  20   00156
20  55  50  43  20  20  20  20  20  20  20  4F  2F  49  20   00160          .ASCII  \ I/O       CPU       Page flts Ph.Mem\<0>    ;
73  74  6C  66  20  65  67  61  50  20  20  20  20  20  20   0016F
                         00  6D  65  4D  2E  68  50  20      0017E
                                                00  00       00186          .ASCII  <0><0>
                                                 010E004D.   00188 P.ABR:  .LONG   17694797
                                                 00000000'   0018C          .ADDRESS P.ABS
                             4B  4E  55  57  52  05          00190 P.ABT:  .ASCII  <5>\RWUNK\
                                     4B  4E  55  03          00196 P.ABU:  .ASCII  <3>\UNK\
                                             53  01          0019A P.ABV:  .ASCII  <1>\S\
                                             4E  01          0019C P.ABW:  .ASCII  <1>\N\
                                             42  01          0019E P.ABX:  .ASCII  <1>\B\
                                             20  01          001A0 P.ABY:  .ASCII  <1>\ \
                                                             001A2          .BLKB   2
43  41  35  21  20  46  41  35  31  21  20  4C  58  38  21   001A4 P.ACA:  .ASCII  \!8XL !15AF !5AC  !3UB!9UL!AS !9UL  !5UW \   ;
21  20  53  41  21  4C  55  39  21  42  55  33  21  20  20   001B3
                    20  57  55  35  21  20  20  4C  55  39   001C2
```

```
                                                                   L 15
SHOWSYSTEM                                                         16-Sep-1984 01:22:08    VAX-11 Bliss-32 V4.0-742              Page  19
V04-000                                                            14-Sep-1984 12:09:48    [CLIUTL.SRC]SHOWSYS.B32;1                   (7)

20  20  20  20  20  20  20  20  20  2F  21  43  41  21  20  001CC            .ASCII  \ !AC!/         !%I\<0><0>                      :
                                        00  00  49  25  21  001DB                                                                    :
                                                  010E003A  001E0 P.ABZ:    .LONG   17694778                                         :
                                                  00000000' 001E4           .ADDRESS P.ACA                                           :
43  41  35  21  20  46  41  35  31  21  20  4C  58  38  21  001E8 P.ACC:    .ASCII  \!8XL !15AF !5AC  !3UB!9UL!AS !9UL  !5UW \      :
21  20  53  41  21  4C  55  39  21  42  55  33  21  20  20  001F7                                                                    :
                    20  57  55  35  21  20  20  4C  55  39  00206                                                                    :
                                            43  41  21  20  00210           .ASCII  \ !AC\                                           :
                                                  010E002C  00214 P.ACB:    .LONG   17694764                                         :
                                                  00000000' 00218           .ADDRESS P.ACC                                           :
43  41  35  21  20  46  41  35  31  21  20  4C  58  38  21  0021C P.ACE:    .ASCII  \!8XL !15AF !5AC  !3UB!3(+)         --  swa\     :
20  20  20  20  29  2B  28  33  21  42  55  33  21  20  20  0022B                                                                    :
                    61  77  73  20  20  2D  2D  20  20  20  0023A                                                                    :
20  20  2D  2D  20  20  74  75  6F  20  20  64  65  70  70  00244           .ASCII  \pped  out  --            !5UW  !AC!/       \    :
21  20  20  57  55  35  21  20  20  20  20  20  20  20  20  00253                                                                    :
                    20  20  20  20  20  20  2F  21  43  41  00262                                                                    :
                            00  00  49  25  21  20  20  20  0026C           .ASCII  \    !%I\<0><0>                                  :
                                                  010E0056  00274 P.ACD:    .LONG   17694806                                         :
                                                  00000000' 00278           .ADDRESS P.ACE                                           :
43  41  35  21  20  46  41  35  31  21  20  4C  58  38  21  0027C P.ACG:    .ASCII  \!8XL !15AF !5AC  !3UB!3(+)         --  swa\     :
20  20  20  20  29  2B  28  33  21  42  55  33  21  20  20  0028B                                                                    :
                    61  77  73  20  20  2D  2D  20  20  20  0029A                                                                    :
20  20  2D  2D  20  20  74  75  6F  20  20  64  65  70  70  002A4           .ASCII  \pped  out  --            !5UW  !AC\             :
21  20  20  57  55  35  21  20  20  20  20  20  20  20  20  002B3                                                                    :
                                                    43  41  002C2                                                                    :
                                                  010E0048  002C4 P.ACF:    .LONG   17694792                                         :
                                                  00000000' 002C8           .ADDRESS P.ACG                                           :

                                                                  SD_LNM$SYSTEM=      P.ABK
                                                                            .EXTRN  SYS$ASCTIM, SYS$TRNLNM

                                                                            .PSECT  $CODE$,NOWRT,2

                                                       007C 00000 PRINT_DATA:
                                                                            .WORD   Save R2,R3,R4,R5,R6                              ; 0463
                                    56 00000000G  00  9E 00002              MOVAB   SHOW$WRITE_LINE, R6                              :
                                    55 00000000G  00  9E 00009              MOVAB   SYS$ASCTIM, R5                                   :
                                    5E        A0  AE  9E 00010              MOVAB   -96(SP), SP                                      :
                      52       04   BC 00000040   8F  C1 00014              ADDL3   #64, @DATA, SCRATCH                              ; 0508
                                                  62  D5 0001D              TSTL    (SCRATCH)                                        ; 0513
                                                  01  12 0001F              BNEQ    1$                                               :
                                                      04 00021              RET                                                      :
     58   AE              00 FF676980   8F 00000000G  00  7A 00022 1$:      EMUL    EXE$GL_ABSTIM, #-10000000, #0, TIME              ; 0521
                                 50     AE        40  8F  9A 00030              MOVZBL  #64, DESC                                        ; 0522
                                 54     AE        04  BC  D0 00035              MOVL    @DATA, DESC+4                                    ; 0523
                                                      7E  D4 0003A              CLRL    -(SP)                                            ; 0527
                                          5C      AE  9F 0003C              PUSHAB  TIME                                             :
                                          58      AE  9F 0003F              PUSHAB  DESC                                             :
                                          5C      AE  9F 00042              PUSHAB  DESC                                             :
                                    65            04  FB 00045              CALLS   #4, SYS$ASCTIM                                   :
                                    54            50  D0 00048              MOVL    R0, STATUS                                       :
                                    03            54  E8 0004B              BLBS    STATUS, 2$                                       :
                                                014B  31 0004E              BRW     19$                                              :
                                 18     AE        10  D0 00051 2$:          MOVL    #16, DESC_SYSNODE                                ; 0536
                                 1C     AE        20  AE  9E 00055          MOVAB   SYSNODEBUF, DESC_SYSNODE+4                       ; 0537
                                        50        08  AE  9E 0005A          MOVAB   TRNLNMLST, $$ITMBLKPTR                           ; 0547
                                        80 00020010   8F  D0 0005E          MOVL    #131088, ($$ITMBLKPTR)+                          :
```

```
                  80        20   AE   9E 00065         MOVAB   SYSNODEBUF, ($$ITMBLKPTR)+                 :
                  80        18   AE   9E 00069         MOVAB   DESC_SYSNODE, ($$ITMBLKPTR)+               :
                                 80   D4 0006D         CLRL    ($$ITMBLKPTR)+                             :
                            08   AE   9F 0006F         PUSHAB  TRNLNMLST                                  : 0557
             08   AE             01   D0 00072         MOVL    #1, 8(SP)                                  :
                            08   AE   9F 00076         PUSHAB  8(SP)                                      :
                         0000'   CF   9F 00079         PUSHAB  P.ABM                                      :
                         0000'   CF   9F 0007D         PUSHAB  SD_LNM$SYSTEM                              :
             10   AE 02000000    8F   D0 00081         MOVL    #33554432, 16(SP)                          :
                            10   AE   9F 00089         PUSHAB  16(SP)                                     :
      00000000G   00             05   FB 0008C         CALLS   #5, SYS$TRNLNM                             :
                  54             50   D0 00093         MOVL    R0, STATUS                                 :
                  05             54   E8 00096         BLBS    STATUS, 3$                                 :
                            18   AE   B4 00099         CLRW    DESC_SYSNODE                               : 0559
                                 27   11 0009C         BRB     8$                                         :
             5F   8F        20   AE   91 0009E 3$:     CMPB    SYSNODEBUF, #95                            : 0565
                                 06   12 000A3         BNEQ    4$                                         :
                            18   AE   B7 000A5         DECW    DESC_SYSNODE                               : 0568
                            1C   AE   D6 000A8         INCL    DESC_SYSNODE+4                             : 0569
                                 53   D4 000AB 4$:     CLRL    I                                          : 0572
 1C   BE      18  AE             3A   3A 000AD 5$:     LOCC    #58, DESC_SYSNODE, @DESC_SYSNODE+4         : 0573
                                 02   12 000B3         BNEQ    6$                                         :
                                 51   D4 000B5         CLRL    R1                                         :
                                 51   D5 000B7 6$:     TSTL    R1                                         : 0574
                                 03   13 000B9         BEQL    7$                                         :
                            18   AE   B7 000BB         DECW    DESC_SYSNODE                               : 0575
                                 53   D6 000BE 7$:     INCL    I                                          : 0573
                  01             53   D1 000C0         CMPL    I, #1                                      :
                                 E8   1B 000C3         BLEQU   5$                                         :
             30   AE     0000'   CF   9E 000C5 8$:     MOVAB   P.ABO, ARGLIST                             : 0582
             34   AE        18   AE   9E 000CB         MOVAB   DESC_SYSNODE, ARGLIST+4                    : 0583
                            38   AE   D4 000D0         CLRL    ARGLIST+8                                  : 0584
             3C   AE        50   AE   9E 000D3         MOVAB   DESC, ARGLIST+12                           : 0585
             50   AE             03   C2 000D8         SUBL2   #3, DESC                                   : 0586
                                 7E   D4 000DC         CLRL    -(SP)                                      : 0591
                         0000'   CF   9F 000DE         PUSHAB  P.ABR                                      : 0592
                            38   AE   9F 000E2         PUSHAB  ARGLIST                                    : 0591
                         0000'   CF   9F 000E5         PUSHAB  P.ABP                                      :
                  66             04   FB 000E9         CALLS   #4, SHOW$WRITE_LINE                        :
                                 62   D5 000EC 9$:     TSTL    (SCRATCH)                                  : 0603
                                 01   12 000EE         BNEQ    10$                                        :
                                      04 000F0         RET                                                :
                  51        0C   A2   9E 000F1 10$:    MOVAB   12(SCRATCH), R1                            : 0609
                                 61   D5 000F5         TSTL    (R1)                                       :
                                 35   15 000F7         BLEQ    13$                                        :
                  0E             61   D1 000F9         CMPL    (R1), #14                                  : 0610
                                 30   14 000FC         BGTR    13$                                        :
                  02             61   D1 000FE         CMPL    (R1), #2                                   : 0613
                                 20   12 00101         BNEQ    12$                                        :
                            2C   A2   D5 00103         TSTL    44(SCRATCH)                                : 0614
                                 1B   19 00106         BLSS    12$                                        :
                                 12   15 00108         BLEQ    11$                                        : 0617
                  0E        2C   A2   D1 0010A         CMPL    44(SCRATCH), #14                           : 0618
                                 0C   14 0010E         BGTR    11$                                        :
                  50        2C   A2   D0 00110         MOVL    44(SCRATCH), R0                            : 0619
                  61     0000'CF 40   D0 00114         MOVL    RSN_TABLE-4[R0], (R1)                      :
                                 17   11 0011A         BRB     14$                                        :
                                   61   0000'   CF 9E 0011C 11$:    MOVAB   P.ABT, (R1)                             ; 0620
                                                10 11 00121         BRB     14$                                     ; 0613
                                   50           61 D0 00123 12$:    MOVL    (R1), R0                                ; 0622
                                   61    0000'CF40 D0 00126         MOVL    STATE_TABLE-4[R0], (R1)                 ;
                                                05 11 0012C         BRB     14$                                     ; 0609
                                   61   0000'   CF 9E 0012E 13$:    MOVAB   P.ABU, (R1)                             ; 0624
                                   50      24   A2 9E 00133 14$:    MOVAB   36(SCRATCH), R0                         ; 0630
                                   54           60 D0 00137         MOVL    (R0), STATUS                            ;
                                           04   A2 D5 0013A         TSTL    4(SCRATCH)                              ; 0631
                                                07 13 0013D         BEQL    15$                                     ;
                                   60   0000'   CF 9E 0013F         MOVAB   P.ABV, (R0)                             ; 0632
                                                1B 11 00144         BRB     18$                                     ;
                   07              60           15 E1 00146 15$:    BBC     #21, (R0), 16$                          ; 0633
                                   60   0000'   CF 9E 0014A         MOVAB   P.ABW, (R0)                             ; 0634
                                                10 11 0014F         BRB     18$                                     ;
                   07              60           0E E1 00151 16$:    BBC     #14, (R0), 17$                          ; 0635
                                   60   0000'   CF 9E 00155         MOVAB   P.ABX, (R0)                             ; 0636
                                                05 11 0015A         BRB     18$                                     ;
                                   60   0000'   CF 9E 0015C 17$:    MOVAB   P.ABY, (R0)                             ; 0637
              10   A2              1F      10   A2 C3 00161 18$:    SUBL3   16(SCRATCH), #31, 16(SCRATCH)           ; 0642
                              04   A2      30   A2 9A 00167         MOVZBL  48(SCRATCH), 4(SCRATCH)                 ; 0651
                              08   A2      31   A2 9E 0016C         MOVAB   49(R2), 8(SCRATCH)                      ; 0652
                                   4F           54 E9 00171         BLBC    STATUS, 22$                             ; 0658
   58   AE         00 FFFE7960     8F      18   A2 7A 00174         EMUL    24(SCRATCH), #-100000, #0, TIME         ; 0661
                              50   AE           10 D0 0017F         MOVL    #16, DESC                               ; 0662
                              54   AE      40   AE 9E 00183         MOVAB   PROCTIM, DESC+4                         ; 0663
                                                7E D4 00188         CLRL    -(SP)                                   ; 0668
                                           5C   AE 9F 0018A         PUSHAB  TIME                                    ;
                                           58   AE 9F 0018D         PUSHAB  DESC                                    ;
                                           5C   AE 9F 00190         PUSHAB  DESC                                    ;
                                   65           04 FB 00193         CALLS   #4, SYS$ASCTIM                          ;
                                   54           50 D0 00196         MOVL    R0, STATUS                              ;
                                   0A           54 E8 00199         BLBS    STATUS, 20$                             ;
                                                54 DD 0019C 19$:    PUSHL   STATUS                                  ; 0671
                       00000000G   00           01 FB 0019E         CALLS   #1, LIB$SIGNAL                          ;
                                                   04 001A5         RET                                             ; 0670
                              18   A2      50   AE 9E 001A6 20$:    MOVAB   DESC, 24(SCRATCH)                       ; 0675
                                   15           54 E9 001AB         BLBC    STATUS, 22$                             ; 0680
                   08         08   BC           04 E1 001AE         BBC     #4, @FLAGS, 21$                         ; 0683
                                                52 DD 001B3         PUSHL   SCRATCH                                 ; 0685
                                        0000'   CF 9F 001B5         PUSHAB  P.ABZ                                   ; 0684
                                                1B 11 001B9         BRB     24$                                     ;
                                                52 DD 001BB 21$:    PUSHL   SCRATCH                                 ; 0687
                                        0000'   CF 9F 001BD         PUSHAB  P.ACB                                   ; 0686
                                                13 11 001C1         BRB     24$                                     ;
                   08         08   BC           04 E1 001C3 22$:    BBC     #4, @FLAGS, 23$                         ; 0691
                                                52 DD 001C8         PUSHL   SCRATCH                                 ; 0693
                                        0000'   CF 9F 001CA         PUSHAB  P.ACD                                   ; 0692
                                                06 11 001CE         BRB     24$                                     ;
                                                52 DD 001D0 23$:    PUSHL   SCRATCH                                 ; 0695
                                        0000'   CF 9F 001D2         PUSHAB  P.ACF                                   ; 0694
                                   66           02 FB 001D6 24$:    CALLS   #2, SHOW$WRITE_LINE                     ;
                                   52      40   A2 9E 001D9         MOVAB   64(R2), SCRATCH                         ; 0700
                                              FF0C 31 001DD         BRW     9$                                      ; 0603
                                                   04 001E0         RET                                             ; 0704

; Routine Size:  481 bytes,    Routine Base:  $CODE$ + 01D4
```

```
; 614          0705  1 END
; 615          0706  0 ELUDOM


                                                        .EXTRN  LIB$SIGNAL, LIB$STOP

;                          PSECT SUMMARY
;
;     Name                        Bytes                    Attributes
;
;  $PLIT$                            716  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $OWN$                             112  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                            949  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;                    Library Statistics
;
;                                         -------- Symbols --------      Pages      Processing
;     File                                Total   Loaded   Percent      Mapped     Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1          18619       62         0      1000         00:01.9



;                          COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:SHOWSYS/OBJ=OBJ$:SHOWSYS MSRC$:SHOWSYS/UPDATE=(ENH$:SHOWSYS)

; Size:          947 code + 830 data bytes
; Run Time:         00:22.7
; Elapsed Time:     01:14.4
; Lines/CPU Min:     1862
; Lexemes/CPU-Min: 23509
; Memory Used:  204 pages
; Compilation Complete
```

0057 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

SHOWTERM
LIS

SHOWMISC
LIS

SHOWPROC
LIS

SHOWSYS
LIS

SHOWMAIN
LIS

SHOWMSCP
LIS

SHOWQUE
LIS

SHOWMSG
LIS

[illegible]